

# A Novel Approach for Robust Detection of Heart Beats in Multimodal Data using Neural Networks and Boosted Trees

Sachin Vernekar<sup>1</sup>, Deepu Vijaysenan<sup>1</sup>, Rohit Ranjan<sup>2</sup>

<sup>1</sup> NIT Surathkal, Surathkal, India

<sup>2</sup> Simon Fraser University, British Columbia, Canada

## Abstract

*This work describes a novel approach designed for Physionet 2014 Challenge, Robust Detection of Heart Beats in Multimodal Data [5]. The objective here is to detect the location of R peaks from QRS complex of an electrocardiogram (ECG) excerpt. Robust detection of heart beats in a noisy ECG signal is an extremely difficult task. To overcome the challenge in such situations, besides ECG, blood pressure (BP) signal is also recorded at the same time; hence the idea here is that, if a segment of one of the signals is noisy, the peaks in that segment can be better estimated by peaks found in the corresponding segment of other signal, if good.*

*The approach uses Machine Learning (ML) methods to identify locations of R-peaks in a given segment of ECG or BP signal. Peaks from both ECG and BP signal are found separately using a novel feature representation and subsequent ML approaches that renders R peaks in the signal, easier to be detected, by a simple windowing technique. Individually detected peaks, from both ECG and BP are further analyzed in chunks of equal short time periods, and the best result of the two is chosen in final peak prediction based on variance comparison techniques. The performance of system on the training dataset [11] provided in the competition is 99.95%. The performance on test datasets which are hidden for phase I, phase II and phase III of the competition respectively are 93.27%, 90.28% and 89.74%. The submission resulted in 1<sup>st</sup> place in all three phases of the competition.*

## 1. Introduction

Manually annotating heartbeat peaks is a cumbersome task, and many algorithms have been proposed to locate the peak locations automatically. There are many scholarly papers which use different approaches such as, filter banks, Doppler radars, Support Vector Machine based detections etc., and the results have been very good on certain standard ECG datasets, favorably because, ECG signals under

consideration are corrupted by presence of known noise patterns, such as noise due to baseline wandering, noise introduced by the sensors, power line noises etc., whose effects can be nullified with suitable filter designs, allowing peak detection systems to be not so robust.

But when few segments of the ECG/BP signal are significantly corrupted by a random noise, making it impossible to detect the location of peaks in those segments, knowledge of peaks located in corresponding segments of the other signal, come handy to robustly detect peaks. This forms the primary objective of the competition; but our solution not only addresses this but also outdoes some of the best solutions in R-peak detection using only ECG (Possibly BP).



Figure 1. Top: Noisy ECG Signal, Bottom: Clear BP Signal.

An example situation is given for this in the Figure 1, where it is shown that a portion of ECG is corrupted by noise and peaks in that segment needs to be replaced with peaks located from the corresponding BP signal.

Quadratic spline wavelet filter [1–4], which is quite extensively used for preprocessing the ECG signal, is used on both ECG and BP signals to remove high frequency noise, as well as low frequency artifacts such as baseline wandering. Two of the four level wavelet decomposition of signals are used to create features to train artificial neural networks (ANN) or boosted tree classifiers for both the signals independently, and peaks are determined from the output of the classifiers, and then the best of the two signal peaks are selected based on short-time variance comparison techniques.

## 2. ECG-BP peak detection

This section gives a brief introduction to data used, pre-processing procedures employed to remove noise, along with a comprehensive step by step procedure to obtain peaks independently from both ECG and BP.

### 2.1. Preprocessing

Both training set [11] and test set contain ECG, BP or both signals digitized at varying frequencies ranging from 120 to 1000 samples per second. The first step in pre-processing involves resampling of these signal to 250 Hz. They are then normalized, so as to have a zero mean and unit variance. They are further passed through a quadratic spline filter [1–3] to filter out noise.

### 2.2. Quadratic Spline filter

This filtering involves passing an ECG/BP signal through a series of low pass and high pass filters as shown in the Figure 2.

The signals out D1, D2, D3 and D4 are called the 1<sup>st</sup> detail, 2<sup>nd</sup> detail etc., respectively. The first low pass filter removes the high frequency noise and the last high pass filter eliminates the baseline wandering problem, which is a low frequency noise. We consider only D4 and D3 details for training the 2-class classifier using an artificial neural network. A representation of D4 for both ECG and BP is shown in Figure 3. As the input signals are of 250 Hz, D4 will have frequency components in the range 15.625 Hz to 31.25 Hz and D3 in 31.25 Hz to 62.5 Hz.

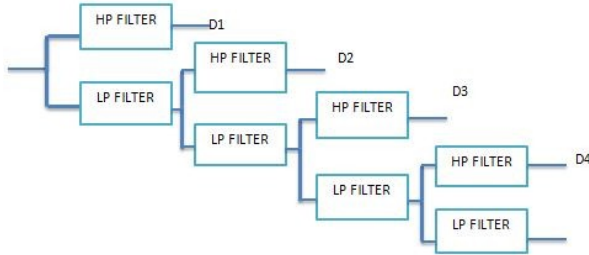


Figure 2. Quadratic Spline Filter.

The coefficients for high pass and low pass filters, and the scaling factor, lambda, for each detail respectively are:

$$H = [0.125, 0.375, 0.375, 0.125]$$

$$L = [2, 2]$$

$$\lambda = [1.5, 1.12, 1.03, 1.01, 1.00]$$

Below is a matlab snippet to find the details:

```
function [wavelets] = findDetails(S)
H = [0.125 0.375 0.375 0.125];
L = [-2 2];
lambda = [1.5 1.12 1.03 1.01 1.00];
numOfDetails = 4;
j = 1;
wavelets=zeros(numOfDetails,size(S,1));
while(j <=numOfStages)
    wavelets(j,:) = conv(S,L,'same')/lambda(j);
    S = conv(S, H, 'same');
    j = j +1;
end
end
```

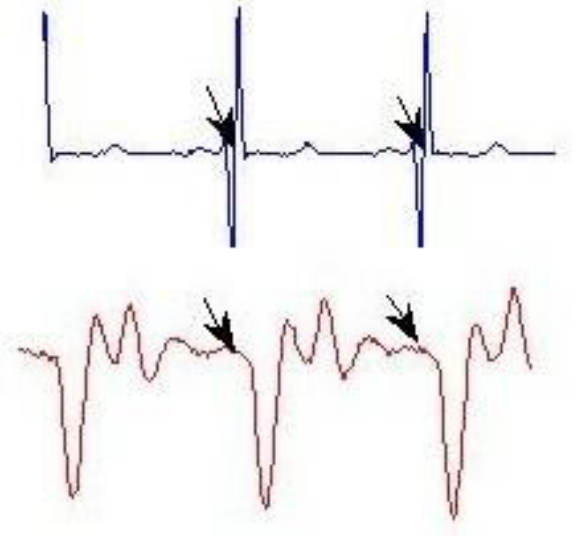


Figure 3. Wavelet Transformed Signals, Top: D4 for ECG, Bottom: D4 for BP. Arrow indicates location of R-peak.

### 2.3. Training

For training neural network, we used the training set provided by the physionet challenge 2014, which contains 100 10-minute long excerpts (records) of multi-parameter recordings of human adults, including patients with a wide range of problems as well as healthy volunteers containing both ECG and BP signals. Each record contains the annotations for R-peak locations.

#### 2.3.1. Training on ECG

The feature size of training data is 202, 101 of which come from D4 and 101 from D3. To choose these features, the procedure employed is as follows:

To construct 1 sample (row) of training set (training matrix), we randomly pick a sample from D4 detail of a signal

(record) called pivot, then take values of D4, 50 samples to the left of the pivot and 50 samples to the right of pivot. These 100 samples plus the value of D4 at pivot constitute 101 features. Similarly we take 101 features from D3 for the same pivot; Thus we get a total of 202 features. Now to label the sample (row), if the pivot happens to be location of an R-peak, we label it 1, else it is a 0. For example, if 77th sample is a pivot and it happens to be an R-peak location, then a row of training matrix containing 202 features and label in Matlab will be as follows,

$$[D4(17 : 126), D3(17 : 126), 1]$$

To get a balanced mix of both classes in the training data, we pick all the R-peak locations across all records to be pivots to form class 1 samples. To get class 0 samples, we choose equal number of samples from each record. While choosing 0s, we make sure, that the pivot doesn't lie in close vicinity of an R-peak location, hence we impose a minimum offset of 20 samples for 0s from location of R-peaks on either sides of it. Our final training data comprised of about 76000 samples each for the 2 classes.

For the competition, we used Matlab/Ocatave Deep Learning Toolbox [8] to train a single hidden layer neural network, with sigmoid activation and dropout functionality and found the optimum value for number of hidden neurons to be 100 with an overall 4-fold training accuracy of close to 94.%.

### 2.3.2. Training on BP

Training for BP is very similar to ECG. Here, instead of selecting a continuous set of 101 features each from both D4 and D3, we select them in an alternate fashion. This is because, we found taking continuous samples had a lot of redundancy and we wanted to model BP in a way slightly different from ECG.

For example, if 102<sup>nd</sup> sample is a pivot, and it happens to be an R-peak location, then a row of training matrix, containing 202 features and label in Matlab will be as follows,

$$[D4(2 : 2 : 202), D3(2 : 2 : 202), 1]$$

When trained over a single hidden layer neural network with sigmoid activation function, 150 hidden neurons and drop-out, we obtained a 4-fold classification accuracy, close to 93%.

### 2.4. Finding Peaks from ANN output

Now the main task is to detect peaks from both ECG and BP signals independently. Since the ANN is trained over input signals of frequency 250Hz, every signal needs to be re-sampled to this frequency. Next step is to normalize the

signals and apply quadratic spline filter and take out D4 and D3 details. Each sample in ECG/BP is taken to be the pivot, and features are extracted as explained in the training section. These are then classified using neural networks trained earlier. This, then gives us a series (vector) of classification probabilities for the signal under consideration, and each of the probability corresponds to the probability of that pivot being an R-peak. The length of ANN output vector is the same as that of the input signal. An example output of ANN is shown in Figure 4 for ECG and similar responses are obtained for BP as well. For further usage, We call this response, peaks-cluster (PC).

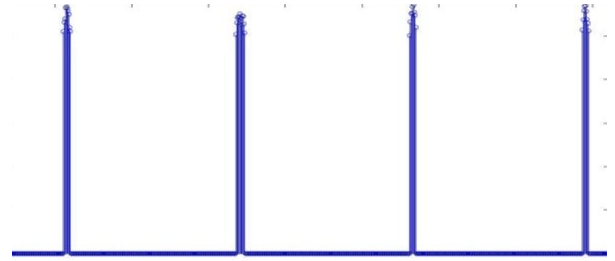


Figure 4. Neural network output for ECG which indicate probability of abscissa being a peak.

The actual R-peaks now correspond to the local maximas in PC, and detecting such local maximas from PC is a much easier task compared to detection from original ECG or BP. To detect peaks from PC, we use a windowing technique; where we use a window of length 60 and slide it along the length of PC to find the local maxima in that region which roughly corresponds to actual R-peak. In the next stage of post processing, we increase the window size based on the average difference calculated between the peaks from the previous results which further eliminate few of the wrongly detected peaks. Finally, some simple thresholding is also used to remove extra peaks, which are a result of P-peaks being comparable to R-peaks in the signal. All these procedures are separately done on both BP and ECG; and resulting two sets of peaks are further examined to take the best result out of them.

### 2.5. Combining Results

We use short-time variance comparison technique, where we take all the peaks corresponding to a certain length of input signal, in our case 5000 samples from both BP and ECG and find the variance of the distance between the peaks and the set of peaks with minimum variance is chosen for the final peak set. Extra care is also taken in order to avoid wrong results occurring from the variance technique, and these are not significant to be discussed and hence can be referred from the link to the code referenced

[12].

Following up at the end of the competition, XGBoost [7] library, which implements gradient boosted trees (extensively used in Kaggle [13] competitions) was used to train the classifiers. The results even though couldnt be obtained for competition dataset, did perform extremely well on the extended dataset [10] released by Physionet [5] which contains relatively noisier signals, results of which are tabulated in Table 1.

### 3. Results

A peak detected is scored as correct, if it is detected within a 150ms window on both sides of the actual R-peak. The scores are reported as per the below scoring function.

$$\text{Sensitivity, } Se = \frac{100 * TP}{TP + FP}$$

$$\text{Positive predictivity, } +P = \frac{100 * TN}{TN + FN}$$

where TP, TN, FP, FN stand for true positive, true negative, false positive and false negative respectively.

$$\text{score} = \frac{Se + (+P)}{2}$$

The results are tabulated in the Table 1.

Table 1. Results of the proposed algorithms on different datasets. (All measurements are in percentages).

Data-set	Only ECG		ECG+BP	
	Se	+P	Se	+P
Using ANN				
MIT/BIH [9]	99.7%	99.8%		
Training set	99.93%	99.92%	99.94	99.96%
Phase III			91.63%	87.86%
Extended set			95.45%	92.24%
Using XGBoost				
Training set			99.95%	99.96%
Extended set			96.48%	92.34%
MIT/BIH	99.75%	99.86%		

The algorithm is found to perform well, even when random noises are deliberately introduced in one of the signals, in which case, all the peaks in the noisy region would be missing and in the final peaks output will be filled by the peaks found from the other signal.

The algorithm also does well in situations where there are apparently double peaks in unusual BP signals, a situation of which is shown in the Figure 5.

The relatively lower results for the hidden datasets in the challenge, may not be the right indicator of the algorithm's performance, as there were possibilities of program getting crashed due some unidentified bugs in the code.

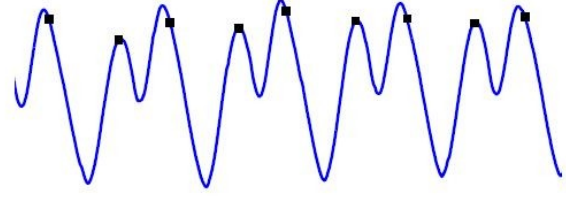


Figure 5. BP signal with significant 2<sup>nd</sup> peak.

### 4. Conclusion

The proposed algorithm gives a better approach, in combining the R-peaks obtained by different physiological signals measured at the same time, so as to find peaks from unrecoverable signal segments corrupted by noise. The algorithm also proposes a novel solution to find R-peaks separately from both ECG and BP signals with very good accuracy on a most of the publicly available standard datasets [9–11]. The results are found to have been relatively better using XGBoost. Since training methods ANN/XGBoost that have been employed are relatively simple, having used only a single classifier, the results may further be improved by ensembling. Using more data and carefully choosing the right samples, avoiding noisy samples in training, may also help.

### References

- [1] Stephane Mallat and sifen Zhong, IEEE Transactions on pattern analysis and machine intelligence, VOL. 14 NO, 7, July 1992.
- [2] Zero-crossings of a wavelet transform, IEEE trans. Inform. Theory, vol. 37, pp, 1019-1033, July 1991.
- [3] S.Zhong, Edges representation from wavelet transform maxima, Ph.D., thesis New York Univ., Sept, 1990.
- [4] Mallat S, Hwang WL. Singularity detection and processing with wavelets. Information Theory IEEE Transactions on 1992;38(2):617643.
- [5] <https://www.physionet.org/challenge/2014>
- [6] <https://www.physionet.org/topresults>
- [7] <https://github.com/dmlc/xgboost/tree/master/python-package>
- [8] <https://github.com/dmlc/xgboost/tree/master/python-package>
- [9] <https://www.physionet.org/physiobank/database/mitdb/>
- [10] <https://physionet.org/challenge/2014/training.zip>
- [11] <https://physionet.org/physiobank/database/challenge/2014/setting/>
- [12] <https://github.com/sachinvernekar/drambuie>
- [13] <https://www.kaggle.com/sachinvernekar>

Address for correspondence:

Sachin Vernekar

Dept. of Electronics and Communication Engineering, NITK, Surathkal, India

sachinvernekar50@gmail.com